REMARKS

Please reconsider the application in view of the above amendments and the following remarks. Applicant thanks the Examiner for carefully considering this application.

Disposition of Claims

Claims 1-25 are pending in the present patent application. Claims 1, 11, 20, and 23 are independent. The remaining claims depend, either directly or indirectly, from claims 1, 11, 20, and 23.

Claim Amendments

Claims 1, 11, 20, and 23 have been amended to clarify the scope of the invention. Further, claims 2, 12, 21, and 24 have been amended to clarify issues arising from the amendments made to independent claims 1, 11, 20, and 23, respectively. No new matter is added by these amendments as support for the amendments may be found, for example, in paragraphs [0022]-[0024] of the referenced application.

Rejection(s) under 35 U.S.C. § 112

Claims 1-25 stand rejected under 35 U.S.C. § 112 as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the Examiner asserts that the specification fails to give an adequate

description of "anonymous consumer state." To the extent the rejection may apply to amended and original claims 1-25, the rejection is traversed.

The specification clearly describes an anonymous consumer state as the state of the tracing framework after a property file is processed. *See* Referenced Application, paragraph [0030]. More specifically, the specification states that "[i]n one or more embodiments of the invention, the *state* of the tracing framework resulting from creating an anonymous consumer state from the object code loaded/written in the *property file* is the same as [the state] would have been if a user-level program (*e.g.,* a consumer) had defined a tracing operation and communicated it to the tracing framework during normal non-boot tracing." *See id.* In other words, the "anonymous consumer state" is described as identical to the state of the tracing framework after a user-level program has initiated and communicated a tracing operation to the tracing framework. Thus, the anonymous consumer state allows the tracing framework to be executed *while* user-level programs are unable to execute. *See* Referenced Application, paragraphs [0023], [0024]. Further, the specification recites that the anonymous consumer state may be claimed by a consumer, allowing the consumer to access the information obtained by the anonymous consumer state. *See* Referenced Application, paragraph [0032]. In view of the above, the anonymous consumer state is adequately described in the specification.

For at least these reasons, claims 1-25 comply with the requirements of 35 U.S.C. §112. Accordingly, withdrawal of this rejection is requested.

Rejections under 35 U.S.C. § 102

Claims 1-25 stand rejected under 35 U.S.C. § 102 as being anticipated by Tamches, "Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels", University of Wisconsin, 2001 ("Tamches"). To the extent that this rejection may still apply to the amended and original claims, the rejection is respectfully traversed.

Independent amended claim 1 recites a method for tracing an instrumented program. More specifically, the instrumented program is traced using a tracing framework while the system is *booting*, where the system *is unable to execute user-level programs while booting*. Typically, kernel level tracing frameworks are unable to trace an instrumented program while a system is booting because the system is unable to execute a user-level program for initiating the tracing framework. *See* referenced application, paragraph [0009]. However, the specification of the referenced application recites that a property file may be used to store the enabling information for the tracing framework. *See* referenced application, paragraph [0026]. While the system is *rebooting*, the property file may be used to enable the tracing framework , which includes creating an anonymous consumer state. *See* referenced application, paragraph [0028]. The anonymous consumer state of the tracing framework is identical to a state that would have been achieved if a user-level program was able to initiate and communicate a tracing operation to the tracing framework. The tracing framework is then able to trace the instrumented program even though the system is *unable to execute user-level programs* (*i.e.*, tracing the instrumented program while booting). *See* referenced application, paragraph [0030].

Turning to the rejection, "[a] claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference."

Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631 (Fed. Cir. 1987) (emphasis added). Further, "[t]he identical invention must be shown in as complete detail as is contained in the claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989). The Applicant respectfully asserts that Tamches does not expressly or inherently describe each and every element of independent claims 1, 11, 20, and 23.

Amended independent claim 1 is directed to a method for tracing an instrumented program. More specifically, amended independent claim 1 recites, in part, that "the system is *unable to execute a user-level program* while *rebooting*". In contrast, Tamches only discloses that dynamic instrumentation is performed by *user-level* programs. *See* Tamches, page 13. In other words, the KernInst system of Tamches allows applications to instrument the kernel using a *user-level* daemon. *See* Tamches, page 26. However, Tamches fails to disclose that instrumented programs are traced while user-level programs are *unable to execute*. In view of this, it is clear that dynamic instrumentation by *user-level* programs as recited in Tamches is not equivalent to tracing while user-level programs are *unable to execute* as explicitly recited in amended independent claim 1.

Furthermore, Tamches fails to disclose tracing an instrumented program after rebooting the system and during the rebooting process as specified in amended independent claim 1. More specifically, amended independent claim 1 recites, in part, "*rebooting* the system after loading the object code" and "wherein while the system is rebooting... the instrumented program is traced using the enabled tracing framework." In contrast, Tamches only teaches that kernels may be *dynamically* modified during *run-time* after the system is rebooted. *See* Tamches, page 5. More specifically, Tamches teaches that "KernInst is loaded and instruments the kernel entirely at *run-time, without* any need to recompile, *reboot*, or even pause the kernel" (emphasis added). Tamches, page 5.

Thus, Tamches explicitly discloses that KernInst does not require the kernel to be rebooted in order to instrument the kernel. In view of this, it is clear that dynamically splicing a kernel *without rebooting* as recited in Tamches is not equivalent to tracing an instrumented program while the system is *rebooting* as recited in independent claim 1.

In view of the above, Tamches fails to disclose all the limitations of amended independent claim 1. Thus, claim 1 is patentable over Tamches. In addition, independent claims 11, 20, and 23 include at least the same patentable subject matter as claim 1 and, thus, are patentable over Tamches for at least the same reasons as claim 1. Dependent claims 2-10, 12-19, 21-22, and 24-25 are patentable for at least the same reasons as the aforementioned amended independent claims. Accordingly, withdrawal of this rejection is respectfully requested.

Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/352001).

Dated: November 9, 2007

Respectfully submitted,

By /Robert P. Lord/

Robert P. Lord
Registration No.: 46,479
OSHA · LIANG LLP
1221 McKinney St., Suite 2800
Houston, Texas 77010
(713) 228-8600
(713) 228-8778 (Fax)
Attorney for Applicant